슈퍼컴을 이용한 전자빔가속기의 차폐설계

강원구* · 김인수 · 국승한 · 김진규 · 한범수 · 정광영[1] · 강창무[2]

이비테크 (주), [1]공주대학교, [2]한국과학기술정보연구원

Shielding Design of Electron Beam Accelerators Using Supercomputer

Won Gu Kang*, In Soo Kim, Sung Han Kuk, Jin Kyu Kim, Bum Soo Han, Kwang Young Jeong[1] and Chang Mu Kang[2]

EB TECH Co., Ltd. 550 Yongsan-dong Yuseong-gu, Daejeon 305-500, Korea
[1]Kongju National University, College of Engineering, CheonAn 330-717, Korea
[2]Korea Institute of Science and Technology Information, Reseat Program, Daejeon 305-806, Korea

Abstract - The MCNP5 neutron, electron, photon Monte Carlo transport program was installed on the KISTI's SUN Tachyon computer using the parallel programming. Electron beam accelerators were modeled and shielding calculations were performed in order to investigate the reduction of computation time in the supercomputer environment. It was observed that a speedup of 40 to 80 of computation time can be obtained using 64 CPUs compared to an IBM PC.

Key words : Supercomputer, SUN Tachyon, Monte Carlo method, Mobile accelerator, Electron beam accelerator, Shielding design

서 론

방사선을 취급하는 산업과 의학 분야에서 방사선 차폐를 위한 정확한 방사선 거동 예측이 요구되고 있으며, 컴퓨터의 계산 속도가 증가함에 따라 몬테카를로 코드(Monte Carlo code)의 이용이 증가하고 있는 추세이다. 또한, 최근에는 가속기 및 차폐설계를 위하여 3차원 고속 모델링 기술의 최적화 및 모사를 통한 설계 및 운전 개선 등에 대한 기술개발의 필요성이 대두되고 있다.

몬테카를로 코드는 중성자, 전자, 감마선의 물질과의 핵반응을 정확하게 기술하고, 3차원의 기하학적 형태를 실물대로 모델링이 용이할 뿐만 아니라 중성자나 감마선의 심층 투과, 흐름, 공기중 산란, 그리고 선량 산정 등의 차폐계산을 쉽게 해결할 수 있다. 그러나 입자를 추적하여 핵물리 현상을 통계적으로 처리함으로서 정확한 예측을 위해서는 과다한 전산처리시간을 요하는 단점이 있다.

따라서 본 연구에서는 몬테카를로 코드의 계산시간을 단축하고자 핵물리와 원자력분야에서 가장 보급화된 MCNP코드(Monte Carlo N-particle transport code)의 병렬프로그램을 KISTI (Korea Institute of Science and Technology Information)의 Tachyon 슈퍼컴퓨터에 설치하여 계산시간의 단축효과를 확인하고자 하였다. 계산시간 비교를 위하여 두 종류의 전자가속기 차폐모델을 IBM PC

* Corresponding authors: Won Gu Kang, Tel. +82-42-930-7506, Fax. +82-42-930-7500, E-mail. unikang@eb-tech.com

Table 1. SUN Blade 6048

Architecture	Cluster
Processor	AMD Opteron 2.0GH (Barcelona)
Node Number	Calculation Node 188
CPU Core Number	3,008
Maximum Capacity	24.58 tflops
Memory	DDR2/667HMz 6TB
Disc storage	SUN X4500/STK6140
Tape storage	SUN SL8500
Operating system	CenOS 4.6
File system	Luster 1.6.6

와 슈퍼컴퓨터에 각각 적용하였다.

재료 및 방법

1. SUN Tachyon 시스템

MCNP의 설치를 위하여 KISTI 슈퍼컴퓨터 4호기의 1차 시스템인 SUN Tachyon Blade 6048 시스템을 선정하였다. 이 시스템은 최고 성능 24 tflops (tera floating point operation per second)을 제공하며, 세계 슈퍼컴퓨터의 Linpack 벤치마크 성능순위에서 130위에 속한다. SUN Tachyon의 주요 사양은 Table 1과 같다 (KISTI 2009c).

계산노드: 188개의 계산노드는 4개의 SUN Blade6048 랙 (rack)에 장착된다. 각 랙은 4개의 선반 (shelf)으로 구성되며, 각 선반은 12개의 x6420 블레이드 (blade)가 위치한다. 각 블레이드는 AMD (Advanced Micro Devices)사의 Barcelona 쿼드코어 2.0 GHz CPU (Central Processing Unit) 4개와 32 GB (Gigabyte) 메모리, HDD (Hard Disk Drive) 대용의 8 GB CF (Compact Flash) 메모리 드라이브 (memory drive)와 2개의 x8 PCI-e 브리지 (bridge)를 가진다.

네트워크 (Network): 인피니밴드 (Infiniband) 네트워크가 노드간 계산 네트워크 및 파일 I/O 통신에 사용된다. 4x IB DDR을 사용하여 non-blocking IB 네트워크를 구축하였으며 이는 노드 간에 채널당 2.5 GB · sec^{-1} (20 Gbps)를 제공한다.

저장소 (Storage): 스크래치와 홈 파일 시스템의 저장은 19대의 SUN X4500 서버로 구성된다. 이는 11TB의 홈 디렉토리 (/home01)와 두개의 글로벌 스크래치 디렉토리 (54TB의 /work01, 40TB의 /work02)를 제공한다. 스크래치와 홈 디렉토리는 Lustre 파일시스템을 통해 계산노드와 기타 노드에 서비스된다. Veritas NetBackup을 사용하여 주기적으로 홈 디렉토리를 테이프에 저장한다.

SUN Tachyon 시스템은 MPI (Message Passing Interface)와 Open MP (Multi-Processing) 병렬프로그램밍을 위하여

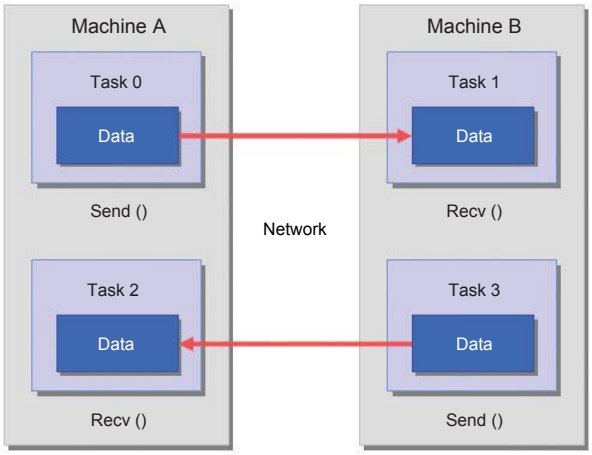

Fig. 1. Memory communication in MPI.

GNU (GNU's not Unix), Portland Group (PGI), Intel이 제공한 컴파일러와 라이브러리 MVAPICH, MVAPICH2, OPEN MPI를 제공한다.

2. 병렬프로그래밍

컴퓨터 프로그램은 순열 (serial)과 병렬 (parallel) 프로그램으로 분류된다. 순열 (serial) 프로그램은 하나의 CPU를 사용하며, 프로그램 명령어 (instruction)를 하나씩 순차로 처리한다. 순열 프로그램은 I/O (input/output)와 같이 느린 프로세스로 인하여 병목 (bottleneck) 현상을 일으키는 단점이 있다.

순열프로그램의 단점을 개선하기위한 방법으로 병렬 (parallel) 프로그램을 사용한다. 병렬 (parallel) 프로그램은 계산을 위하여 여러 개의 CPU를 사용하며, 문제를 여러 개의 task로 분리하여 병렬로 동시 처리한다.

병렬계산은 여러 CPU를 내장한 컴퓨터나 여러 컴퓨터를 연결한 네트워크에서 가능하다 (Barrney 2009a). 예로 KISTI의 Tachyon 시스템은 3008 CPU를 포함하며 24 tflop을 제공한다. 또한 Berkeley의 seti@home network는 330,000개의 컴퓨터와 연결되며 528 tflop을 제공한다.

병렬계산은 순열계산에 비하여 대량 문제를 다루고, 계산속도와 비용을 줄이는 효과가 있다. 병렬계산을 위한 프로그램 방법에는 MPI와 Open MP 등이 사용된다.

1) MPI (Message Passing Interface)

초기의 MPI는 메모리를 공유하지 않는 환경에서 병렬프로그램밍을 위하여 다중 프로세서 간의 데이터 통신의 표준을 제공하였다. MPI는 point-to-point 혹은 집합 프로세서 간의 통신을 지원한다. 통신 모드로는 동기송신, 준비송신, 버퍼송신, 표준송신이 있다. Fig. 1에 MPI의

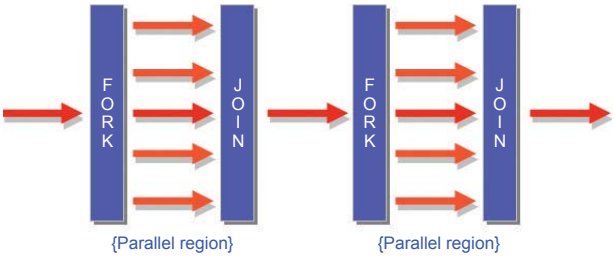

Fig. 2. Fork-Join concept in Open MP.

네트워크를 이용한 프로세스 간의 메모리 통신 기능을 보여주고 있다.

현재의 MPI는 공유 메모리(shared memory) 혹은 MPM (Massively Parallel Machine), SMP (Symmetric Multiple Processors), 워크스테이션 클러스터(workstation clusters), 이기종네트워크(heterogeneous network)와 같은 병렬 구조 시스템에 사용된다(Barrney 2009b; KISTI 2009a).

2) Open MP (Multi-Processing)

Open MP는 공유 메모리 환경에서 병렬화를 위한 표준을 제공하며, 포크(fork)-조인(join) 개념을 적용하고 있다. Open MP의 포크-조인 개념은 초기에 단일 스레드(thread)로 시작하다가, 병렬구조를 만나면 N개의 스레드로 나뉘어 코드를 수행하며, 병렬구조의 마지막에는 다시 단일 마스터 스레드로 합친다(Barrney 2009c; KISTI 2009b). Fig. 2에 Open MP의 포크-조인 개념을 설명하고 있다.

Open MP는 소스 코드를 많이 수정할 필요가 없으며, 특히 DO LOOP와 같이 숫자를 많이 처리하는 부분에서 매우 효과가 있다.

Open MP는 MPI와 함께 사용이 가능하다. 예로 MPI로 여러 프로세서로 분산하고, Open MP로 프로세서의 병렬화 수준을 더욱 향상시키는 방법을 사용한다.

3. 계산속도

병렬프로그램의 속도는 사용하는 프로세스의 수와 프로그램의 병렬화에 의존한다. 예로 p를 병렬화 부분, s=1−p를 순차 부분, N을 사용하는 프로세스로 정의하면, 병렬 프로그램 계산의 가속(speedup)은 아래 식으로 표시된다.

$$\text{Speedup}=\frac{1}{\frac{p}{N}+s}$$

예로, N=10, p=0.8이면,

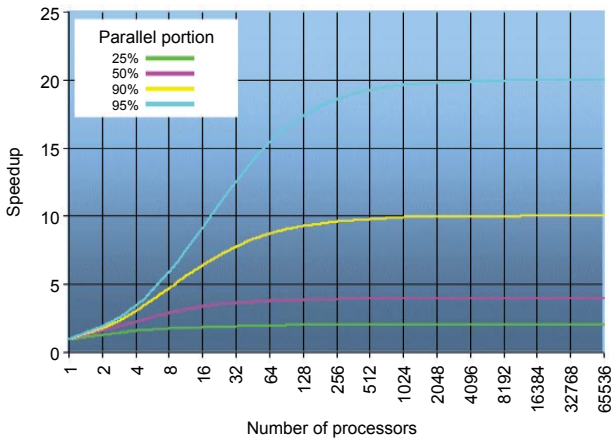

Fig. 3. Number of processors vs. speedup.

$$\text{Speedup}=\frac{1}{\frac{0.8}{10}+0.1}=\frac{1}{0.18}=5.55$$

N=100, p=0.8이면,

$$\text{Speedup}=\frac{1}{\frac{0.8}{100}+0.1}=\frac{1}{0.108}=9.25$$

Fig. 3은 여러 가지의 병렬프로그램과 프로세스 수에 대한 계산의 가속화를 보여주고 있다. 가속은 프로세스 수에 따라 증가하다가 포화되는 경향이 있다. 병렬화가 1.0에 접근하면 가속은 무한대로 증가한다.

4. MCNP5 설치

Tachyon 슈퍼컴에 병렬프로그램 설치를 위하여 MCNP version 5 (Brown 2003)를 사용하였다. MCNP5는 이전의 MCNP4C2 (Briesmeister 2000)를 ANSI 표준 FORTRAN 90으로 전환하였고, 광자-핵 충돌모델, time-splitting, 그래픽 등의 기능을 추가하였다. 또한 MCNP5는 슈퍼컴에서 OPEN MP와 MPI의 평행처리 기능을 위한 설치 파일을 포함하고 있다.

Tachyon에 설치를 위하여 Unix system에서 MCNP 5.1.40 프로그램 패키지의 UNIX CD-ROM에서 .tgz 파일을 un-tar하였다:

$ tar xvfz MCNP5_RSICC_1.40.tgz

이 과정에서 디렉터리(directory) MCNP5와 서브디렉터리(subdirectories) Source, Testing, Manual, Documents, bin, Sample_problems, Pstudy가 생성된다.

MCNP를 설치하기 위하여 프로그램 패키지에 포함된 install을 사용하여 makefile을 작성한다.

- Source 디렉터리로 옮긴 후, ./install를 명령을 수행한다.

  $ ./install

  install을 수행하는 절차는 다음과 같다:
- 상위 메뉴에서 4번을 선택하여 Sequential (Seq), Message Passing Interface (MPI), Open Multi-Processing (Open MP) 중 하나를 선택한다.
- MPI를 선택한 경우 MP 하위 선택 메뉴에서 프로세서 수를 지정한다.
- Open MP를 선택하는 경우 MP 하위 선택 메뉴에서 스레드 (thread) 수를 지정한다.
- 아래의 컴파일러 (compiler) 디렉터리를 지정한다.

  /application/mpi/openmpi/intel/bin/mpirun
- 아래의 그래픽 (graphic) 디렉터리를 지정한다.

  /usr/S11R6/lib64

  /usr/X11R6/lib64/X11
- 상위 메뉴에서 M: makefile을 선택한다.
- X로 Exit한다.

makefile이 작성되면 UNIX의 gmake utility를 사용하여 프로그램을 만든다 (build). build 작업 이전에 작업 중인 디렉터리 (.)와 필요한 컴파일러가 PATH에 속하여야 한다. 또한 핵자료 디렉터리 (directory)와 라이브러리 (library)가 DATA PATH로 위치를 발견할 필요가 있다. gmake에 의하여 생성된 컴파일된 프로그램은 src 디렉터리에 위치한다. gmake 명령은 프로그램의 종류에 따라 아래와 같이 수행한다.

$ gmake clean

$ gmake build CONFIG=seq #sequential program

$ gmake build CONFIG=mpi #mpi program

$ gmake build CONFIG='mpi omp' #mpi+open mp의 혼성

컴파일 (compile)된 mcnp5.mpi는 /Source/src/ 디렉터리에 저장된다.

Table 2에 SUN tachyon에서 생성된 mcnp5 순열과 병렬프로그램을 보여주고 있다.

5. 계 산

Tachyon 시스템에서 계산을 수행하려면 job script를 작성하여 아래와 같이 제출한다.

$ qsub mcnp.sh

여기서 mcnp.sh은 job script 파일로 시스템이 batch job que를 결정하기 위한 파라미터를 지정한다. 예로 mcnp5.mpi 프로그램을 node당 4개의 CPU와 총 16개의 CPU를 사용하여 입력 inp01을 계산하는 경우, mcnp.sh를 Table 3과 같이 정의할 수 있다.

Table 2. mcnp5 parallel programs

mcnp5	serial, mpi, hybrid (mpi+omp)
mcnp5	serial
mcnp4.mpi	4 mpi
mcnp16.mpi	16 mpi
mcnp64.mpi	64 mpi
mcnp4 × 4.mpi	hybrid, 4 mpi+4 thread
mcnp8 × 8.mpi	hybrid, 8 mpi+8 thread
mcnp16 × 4.mpi	hybrid, 16 mpi+4 thread

Table 3. mcnp.sh file

```
#$ -pe mpi_4cpu 16 # 4cpu per a node, total 16 cpu
#$ -q normal
#$ -l h_rt=01:00:00 # maximum 1 hour
#$ -M MyEmailAddress
#$ -m e #sending a email after complete works
Export OMP_NUM_THREADS=4
mpirun -machinefile TMPDIR/machines
        -np $NSLOTS mcnp5.mpi i=inp01
```

Table 4. Differences from reference sample outputs (42 cases)

Input case	OUTP	MCTAL	Remarks
#13	84202	3122	tally volume
#18	126692	16254	lattice cell input
#23	596	0	format
#41	0	147	format
#42	514	0	format

Tachyon 시스템에 설치된 MCNP5 병렬프로그램을 검증하기 위하여 42종류의 샘플입력 (sample input)에 대하여 계산을 수행하며 MCNP5 패키지에 포함된 참고 출력 (OUTP, MCTAL) 파일과 비교하였다. 두 프로그램은 좋은 일치를 보여 주었고, Table 4에서 보는 바와 같이 발견된 차이는 포맷 (format)의 차이가 주종을 이루고, 차폐계산에는 영향을 주지 않는 것으로 나타났다.

MCNP5 병렬프로그램의 계산시간을 비교하기 위하여 이비테크에서 개발한 이동형가속기 (Kang 2009)와 저에너지 가속기의 2가지 모델의 차폐설계를 고려하였다.

몬테카를로 계산을 위하여 두 전자빔 가속기를 구성하는 가속관, 전자빔 인출장치, titanium window, 빔 표적, 차폐벽 등 가속기의 부품을 3차원으로 모델링하였다.

이동형가속기 (Fig. 4)의 측면 차폐는 중앙에 납 7 cm과 양면에 1 cm의 철판을 사용하고, 하부차폐는 5 cm의 납과 양면에 1 cm 철판을 사용하였다 (Kang 2009). 저에너지가속기의 차폐 (Fig. 5)는 측면과 하부에 1.2 cm 철판을 사용하였다.

가속기의 전자빔은 cosine 각분포를 가진 표면 선원을

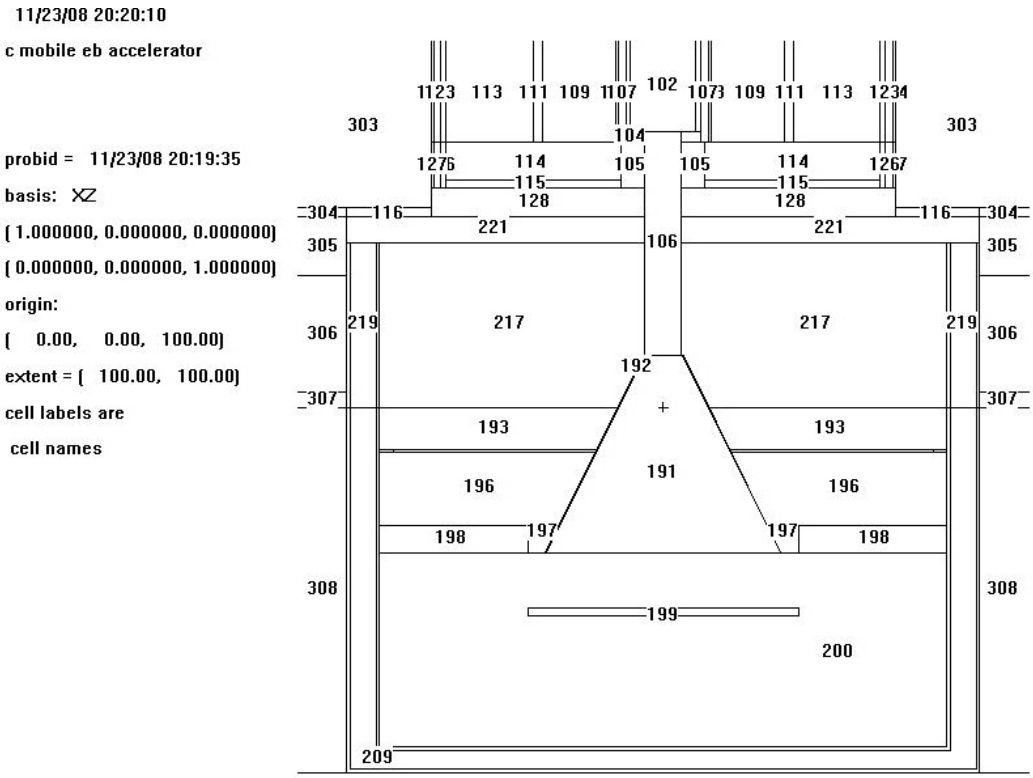

Fig. 4. MCNP model of mobile EB accelerator.

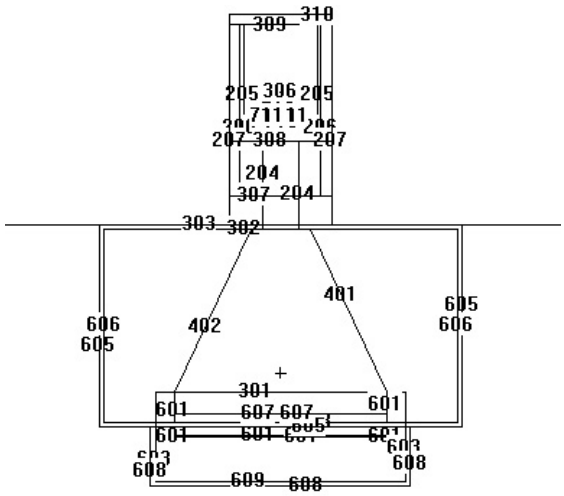

Fig. 5. MCNP model of low energy EB accelerator.

가정하였고, 전자빔의 에너지와 전류는 Table 5와 같다.

결과 및 논의

이동형 및 저에너지 전자빔 가속기에 대한 계산 시간

Table 5. Electron beam sources

	Mobile EB accelerator	Low energy EB accelerator
Beam energy, MeV	0.6	0.075
Beam current, mA	33	20

Table 6. Computation time (hh : mm : ss) - mobile EB accelerator

Method	Histories			
	10^6	10^7	10^8	10^9
IBM PC (1.2 GHz)	00:02:35	00:25:47	04:17:35	
Serial, 1 mpi	00:01:27	00:13:44	02:15:40	
4 mpi	00:00:33	00:04:50	00:49:50	
16 mpi	00:00:15	00:01:03	00:09:29	01:39:14
64 mpi	00:00:12	00:00:45	00:05:48	00:58:05
4 mpi+4 omp	00:00:20	00:01:58	00:19:36	
8 mpi+8 omp	00:00:13	00:00:45	00:05:12	00:56:01
16 mpi+4 omp	00:00:14	00:00:47	00:05:34	00:57:36

이 Table 6과 Table 7에 나타냈으며, 10^6～10^9의 히스토리(history)를 사용하였을 때, 각 병렬프로그램에 소요되는 시간을 비교하였다. 결과적으로 아래와 같이 결론지을 수 있다.

Tachyon 순차프로그램은 IBM PC 순차프로그램에 비

Table 7. Computation time (hh : mm : ss) - Low energy EB accelerator

Method	Histories			
	10^6	10^7	10^8	10^9
IBM PC (1.2 GHz)	00:02:59	00:35:46	04:22:13	
Serial, 1 mpi	00:00:44	00:08:06	01:10:13	
4 mpi	00:00:17	00:02:40	00:26:23	
16 mpi	00:00:09	00:00:34	00:05:01	01:54:15
64 mpi	00:00:11	00:00:27	00:03:21	00:29:52
4 mpi+4 omp	00:00:10	00:01:00	00:09:01	
8 mpi+8 omp	00:00:11	00:00:26	00:03:11	00:27:37
16 mpi+4 omp	00:00:08	00:00:27	00:03:00	00:32:35

하여 계산시간이 2 내지 4배 가속되었다. 이는 두 시스템의 CPU 속도 차이에서 기인한다.

mpi CPU 수의 증가에 따라 계산 시간이 감소하였다. 예를 들어 저에너지 가속기의 경우, 10^8 히스토리에서 4, 16, 64 mpi CPU는 IBM PC에 비하여 계산시간이 각각 10, 50, 80배 가속하였고, 이동가속기의 경우는 각각 5, 25, 40배 가속하였다.

hybrid (mpi+omp)는 mpi만 사용하였을 때 보다 계산시간이 더 단축된 것으로 나타났다. 예를 들어 이동형 가속기의 경우, 10^8 히스토리에서 4 mpi와 4 mpi+4 omp는 각각 49분 50초와 19분 36초로 나타났다.

전체 CPU수가 같은 경우, hybrid와 mpi는 유사한 계산 시간이 소요되었다. 이동형 가속기의 경우, 10^8 히스토리에서 16 mpi와 4 mpi+4 omp는 각각 9분 29초와 19분 36초로 나타났다.

결 론

MCNP5 중성자, 전자, 광자 몬테카를로 프로그램을 병렬프로그래밍으로 KISTI SUN Tachyon 슈퍼컴퓨터에 설치하였다. 슈퍼컴퓨터 환경에서의 계산시간 단축을 위하여 2가지 모델의 전자가속기에 적용하여 차폐계산을 수행하였다.

계산결과 Tachyon 슈퍼컴에서 병렬프로그램으로 64개의 CPU를 사용하는 경우, 방사선 차폐를 위한 계산시간이 IBM PC에 비하여 40～80배까지 가속되었다.

본 연구의 슈퍼컴을 이용한 몬테카를로 기법은 전자빔 가속기, 차폐 설계 및 전자빔공정 모사에서 계산 시간을 대폭적으로 단축할 수 있어 생산성 증가, 품질향상 및 방사선 안전에 기여할 것으로 기대한다. 또한 핵변환, 가속기구동원자로 설계, 신소재 개발에 이용되고, x-ray CT (Computed Tomography) 모사, 동위원소 치료 등에 널리 이용될 수 있다.

사 사

본 과제의 수행을 위하여 슈퍼컴퓨터의 사용과 병렬프로그래밍 설치 등 기술적 도움을 주신 KISTI 슈퍼컴퓨팅센터 (KSC) 및 관련 스태프에 감사를 드립니다.

참 고 문 헌

KISTI. 2009a. MPI를 이용한 병렬 프로그래밍, KISTI 슈퍼컴퓨팅센터.

KISTI. 2009b. Open MPI를 이용한 병렬 프로그래밍, KISTI 슈퍼컴퓨팅 센터

KISTI. 2009c. 슈퍼컴퓨터 (SUN Tachyon System) H/W, S/W 환경 소개 및 실습, KISTI 슈퍼컴퓨팅센터.

Briesmeister JF. Ed. 2000. MCNP-A general monte carlo N-particle transport code. Version 4C. LA-13709. Los Alamos National Laboratory.

Brown FB. 2003. NCNP-A general monte carlo particle transport code. Version 5. Report LA_UR-03-1987 Los Alamos National Laboratory.

Barrney B. 2009a. Introduction to parallel programming. Lawrence Livermore National Laboratory.

Barrney B. 2009b. Message Passing Interface (MPI). Lawrence Livermore National Laboratory.

Barrney B. 2009c. Open MP. Lawrence Livermore National Laboratory.

Kang WG, Kuk SH, Kim JK, Han BS and Kang CM. 2009. Shielding design of a mobile electron beam accelerator using monte carlo technique. *J. of Radiation Industry* **3**(2):79-85.

Manuscript Received: February 9, 2010
Revision Accepted: February 18, 2010